

- Faculté des sciences
- www.unine.ch/sciences

Informatique générale : programmation II (3IN1023)

Filières concernées	Nombre d'heures	Validation	Crédits ECTS
Bachelor en biologie	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6
Bachelor en mathématiques	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6
Bachelor en science des données	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6
Bachelor en systèmes naturels	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6
Master en informatique	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6
Pilier B A - mathématiques	Cours: 2 ph Exercice: 2 ph	Voir ci-dessous	6

ph=période hebdomadaire, pg=période globale, j=jour, dj=demi-jour, h=heure, min=minute

Période d'enseignement:

- Semestre Printemps

Equipe enseignante

Prof. Pascal Felber

Contenu

Le cours expose les principes de la programmation et du développement de logiciels complexes à l'aide de langages à objets. Le cours est complété par des travaux pratiques et le développement de projets.

Le cours utilise les langages programmation Python et Java.

Les principaux thèmes traités sont :

- Principes d'abstraction
- Objets et classes
- Méthodes et invocations
- Groupage d'objets
- Encapsulation, héritage, sous typage, polymorphisme
- Traitement d'erreurs et exceptions
- Interfaces graphiques
- Développement d'applications et études de cas

Forme de l'évaluation

Contrôle continu noté (séries de travaux pratiques notées et évaluation écrite des connaissances de 90 minutes en fin de semestre, comptant respectivement pour 50% et 50% de la note).

Modalités de rattrapage

En cas d'échec, un examen oral de rattrapage de 30 minutes pourra avoir lieu en lieu de l'évaluation écrite des connaissances. A coordonner directement avec le professeur pour une session ultérieure.

Documentation

<https://moodle.unine.ch>

Pré-requis

Programmation I

- Faculté des sciences
- www.unine.ch/sciences

Informatique générale : programmation II (3IN1023)

Forme de l'enseignement

Cours + exercices

Objectifs d'apprentissage

Au terme de la formation l'étudiant-e doit être capable de :

- Formuler un problème informatique
- Définir une solution (algorithme)
- Expliquer un algorithme
- Mettre en oeuvre un programme
- Pratiquer la programmation

Compétences transférables

- Stimuler sa créativité
- Conceptualiser un problème
- Synthétiser une solution